THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data

Citation for published version:
Shkurti, A, Goni, R, Andrio, P, Breitmoser, E, Bethune, I, Orozco, M & Laughton, C 2016, 'pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data', *SoftwareX*.
https://doi.org/10.1016/j.softx.2016.04.002

Digital Object Identifier (DOI):
10.1016/j.softx.2016.04.002

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
SoftwareX

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. May. 2023

ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

SoftwareX

SoftwareX ▮ (▮▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/softx

pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data

Ardita Shkurti[a], Ramon Goni[b,c], Pau Andrio[b,c], Elena Breitmoser[d], Iain Bethune[d], Modesto Orozco[b,c,e,f], Charles A. Laughton[a,*]

[a] *School of Pharmacy and Centre for Biomolecular Sciences, The University of Nottingham, University Park, Nottingham, NG7 2RD, United Kingdom*
[b] *Barcelona Supercomputing Center, Jordi Girona 31, Barcelona 08034, Spain*
[c] *Joint BSCCRG-IRB Program in Computational Biology, Barcelona, Spain*
[d] *Edinburgh Parallel Computing Centre (EPCC), The University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, United Kingdom*
[e] *Institute for Research in Biomedicine (IRB Barcelona), Baldiri Reixach 10-12, 08028 Barcelona, Spain*
[f] *Department of Biochemistry and Molecular Biology, University of Barcelona, 08028 Barcelona, Spain*

Received 29 November 2015; received in revised form 5 April 2016; accepted 7 April 2016

Abstract

The biomolecular simulation community is currently in need of novel and optimised software tools that can analyse and process, in reasonable timescales, the large generated amounts of molecular simulation data. In light of this, we have developed and present here pyPcazip: a suite of software tools for compression and analysis of molecular dynamics (MD) simulation data. The software is compatible with trajectory file formats generated by most contemporary MD engines such as AMBER, CHARMM, GROMACS and NAMD, and is MPI parallelised to permit the efficient processing of very large datasets. pyPcazip is a Unix based open-source software (BSD licenced) written in Python.
© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Keywords: Data analysis; Principal component analysis; Molecular dynamics; Molecular simulation

Software metadata

Current software version	1.4.4
Permanent link to executables of this version	For this Python software the source code can be downloaded at: https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082
Legal Software License	BSD
Computing platform/Operating System	Unix-like
Installation requirements & dependencies	numpy, scipy, setuptools, MDAnalysis, cython, mdtraj, h5py, argparse, cython, netCDF4
If available, link to user manual—if formally published include a reference to the publication in the reference list	https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/pyPCAZIP_manual.docx and https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/pyPCAZIP_manual.pdf
Support email for questions	ardita.shkurti@nottingham.ac.uk

* Corresponding author.
E-mail address: Charles.laughton@nottingham.ac.uk (C.A. Laughton).

http://dx.doi.org/10.1016/j.softx.2016.04.002
2352-7110/© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

Software metadata

Current code version	1.4.4
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082
Legal Code License	BSD
Code versioning system used	git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	g++ compiler, Unix-like OS, and dependencies: numpy, scipy, setuptools, MDAnalysis, cython, mdtraj, h5py, argparse, cython, netCDF4
If available Link to developer documentation/manual	https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/pyPCAZIP_manual.docx and https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/pyPCAZIP_manual.pdf
Support email for questions	ardita.shkurti@nottingham.ac.uk

1. Introduction

Molecular dynamics simulations of biological molecules and complexes can give insights into the relationship between macromolecular structure and dynamics at the atomistic level, and the complex emergent properties of the system at much longer length and timescales. Continuing developments in hardware and software mean that researchers are faced with ever increasing volumes of raw data from the simulations that need to be stored, analysed, and shared. The key raw data are trajectories: snapshots of the time-evolution of the system output at regular intervals, where each snapshot records the three dimensional coordinates of each atom in the simulation at that moment in time. The widening gap between highly scalable molecular simulation codes that enable simulation of multi-million atom systems over microseconds of time, and legacy sequential analysis tools, that were designed to deal with tens to hundreds of thousands of atoms over nanoseconds of time, is exposing a new bottleneck in the process of obtaining scientific insights from the computational experiments.

As one of the efforts to address this gap we have developed pyPcazip, a suite of software tools that can compress molecular simulation data to a small fraction (few percent) of their original size without significant loss of information. According to their interests, users can control the balance between the pyPcazip degree of compression and precision (fraction of the overall variance that is captured from the original molecular simulation data) of the molecular simulation data. Subsequently, the compressed data opens the door to a manifold of analysis methods that produce objective, quantitative and comparative metrics related to convergence and sampling of molecular simulation as well as metrics on the similarity between molecular simulation trajectories.

2. Problems and background

pyPcazip uses Principal Component Analysis (PCA), a dimensionality reduction technique at the core of its algorithms for compression and analysis of MD trajectory data.

Dimensionality reduction techniques such as PCA are increasingly being applied to the analysis of molecular simulation data [1–4], as well as other types of data that report on variations in biomolecular conformation such as NMR ensembles [5], collections of crystal structures [6], and Monte Carlo simulations [7].

PCA allows the dominant modes of molecular flexibility to be identified in a rigorous manner, and presented in the form of variations in the values of a small number of collective coordinates, rather than the 3N independent Cartesian coordinates of the individual atoms, so greatly easing interpretation and visualisation. PCA provides the gateway to a range of analysis methods that provide quantitative and comparative metrics related to convergence and sampling, and the similarity between one trajectory and another.

We have previously described how this method can be applied to compression of the data [8], and as a route to enhanced sampling [9] using our Fortran[1] and C[2] software codes. These software codes which we have developed in previous years, have very limited documentation and a very basic functionality. For these reasons we have undertaken the current code development, re-engineering and substantial functionality enhancement in order to provide the user community with a complete suite of software tools that they can use much more easily, flexibly and compatibly with their needs and that they can cite.

In fact, here and now, with pyPcazip we present a complete new suite of software tools written in Python that includes some redesigned and reengineered algorithms of our Fortran and C codes but a much better engineered software and a much more extended range of functionalities with respect to these formerly used codes of ours. Originalities of pyPcazip (not present in our Fortran and C codes) include but are not limited to: (i) A better handling of memory issues when dealing with very large datasets; (ii) On-the-fly selection of subsets of atoms of interest for the PCA analysis from the available datasets (rather than having the datasets filtered by the user prior to using the software); (iii) Flexible support for the simultaneous analysis of multi-trajectory datasets that vary in their molecular topology and number of atoms; (iv) MPI support for input processing and internal calculations; (v) Compliance with High Performance Computing (HPC) architectures such as ARCHER

[1] http://holmes.cancres.nottingham.ac.uk/pcazip.
[2] http://mmb.pcb.ub.es/software/pcasuite/pcasuite.html.

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

(UK national supercomputing resource); (vi) Unit testing and an automatic testing suite; (vii) Compliance with a large range of state-of-the-art formats (output formats of MD engines such as AMBER [10], CHARMM [11], GROMACS [12] and NAMD [13]) and analysis tools (such as MDAnalysis [14]) of MD code outputs.

3. Software framework

3.1. Software architecture

In our software, the compression of an MD trajectory of N atoms using PCA is implemented through a workflow that involves:

- Calculating the average structure and least-squares fitting each snapshot to this average structure;
- Calculating the Cartesian coordinate covariance matrix of the fitted data;
- Diagonalisation of this covariance matrix to obtain a set of N eigenvectors and eigenvalues;
- Selecting the top M of these that capture a chosen percentage of the total variance (e.g. 90%);
- Calculating, for each snapshot in the trajectory, the corresponding projection into the M-dimensional subspace;
- Writing the selected eigenvectors, eigenvalues, and projections to the output file.

3.2. Software functionalities

As a suite of software tools, pyPcazip is composed of four main components and related functionalities:

- pyPcazip itself takes one or many input MD trajectory files and converts them into a highly compressed, HDF5-based,[3] *.pcz* format. The program has options to select subsets of atoms, and/or subsets of snapshots from the trajectory files for analysis. The file-reading capabilities of pyPcazip draw extensively on the MDAnalysis Python toolkit [14]. In addition to providing pyPczdump (see below) as a tool to post-process the *.pcz* format we also provide a customised reader of the output files produced by pyPcazip as part of a module of the software. This module[4] can be found within the source code and could be easily integrated and used in third-party software if needed upon citation of this work.
- pyPcaunzip can decompress a .pcz file back into a conventional trajectory file in a range of formats (including dcd, .xtc, .ncdf, .trj).
- pyPczdump extracts information such as eigenvectors, eigenvalues, and projections from a .pcz file. It can also produce multi-model PDB format files to animate eigenvectors.
- pyPczcomp permits the quantitative comparison of the data from two congruent .pcz files. An example might be the dynamics of a protein in the presence and absence of a ligand, or a comparative analysis of the dynamics of a wild-type protein and a mutant.

[3] https://www.hdfgroup.org/HDF5/doc/H5.format.html.

[4] https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/ramonbsc-pypcazip-3d7ab553c8dc/pypcazip/MDPlus/analysis/pca/pcz.py.

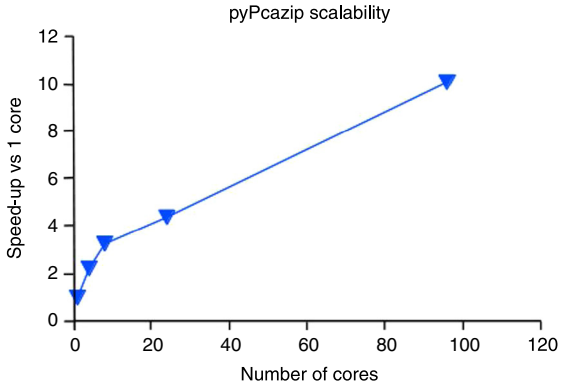

Fig. 1. Parallel scaling performance of pyPcazip.

4. Implementation and empirical results

4.1. Implementation details

pyPcazip is a Python software code that provides command-line tools for the compression and analysis of molecular dynamics trajectory data using PCA methods. The software is designed to be flexible, scalable, and compatible with other Python toolkits that are used in the molecular simulation and analysis field such as MDAnalysis [14].

Many stages in the pyPcazip workflow such as the input reading process and the covariance matrix calculation of the PCA analysis are amenable to parallelisation, which has been implemented using MPI. Fig. 1 shows scaling data of pyPcazip on up to 96 cores on ARCHER-UK national supercomputing service. Each of the ARCHER compute nodes contains two 2.7 GHz, 12-core processors for a total of 24 cores per node. Where the number of cores is less than 24, MPI processes are assigned to the one processor first, then the second, and in the other cases multiples of 24 processes have been used, keeping the nodes fully populated. The bend in this figure at 12 cores occurs as the first processor of the first node has become fully occupied. With less than 12 cores used, each core has access to proportionally more cache space and memory bandwidth, so the higher performance is obtained. The dataset for this scalability analysis includes 10,000 snapshots of a Mouse Major Urinary Protein (MUP) [15] trajectory (with 35k atoms including solvent) that is atom-filtered on-the-fly selecting the backbone related atoms only (157 alpha-carbon atoms).

An automatic testing suite has been developed for pyPcazip that is activated by a single command ("pyPcazip –tests"). This validates the correct functioning of the installed software. In addition, individual python modules are instrumented with extensive unit tests.

Installation instructions, details of underlying algorithms, detailed performance data, and use-case examples are available at https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082.

4.2. Empirical results

The compression achieved by pyPcazip depends on the nature of the molecular system and, as a "lossy" method,

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

on the chosen quality threshold (percentage of the total variance to be captured). In particular, the variance captured considering a small number of the most important principal components (modes) retains crucial insights for conformational investigations as these modes directly relate to the highest amplitude (and normally slower) motions of molecular systems whereas the less important principal components relate to the high frequency small amplitude atomistic fluctuations. For this reason we would not recommend the use of this suite of tools for the investigation of phenomena such as reaction mechanisms where the retention of sub-angstrom accuracy in e.g. bond lengths is required.

Table 1 illustrates performance for three example datasets, each of 1000 snapshots: (1) a short peptide (alanine-12, unpublished data); (2) a DNA 18mer [16]; and (3) the mouse major urinary protein (MUP) [15]. The two metrics are the degree of compression achieved and the compression error expressed as the average RMSD between snapshots in the original file and those in the compressed file. Clearly for many purposes compression to 10%–20% of the original file size is quite possible. The code has been tested on a variety of platforms ranging from laptops to national HPC facilities, and compression/decompression gives results identical to within an RMSD of less than 0.02 Å, even when run in parallel.

5. Illustrative examples

The pyPczdump and pyPczcomp utilities allow a range of PCA-related metrics extracted or calculated from the compressed trajectory files. Output is in the form of ASCII data files that may be easily rendered using the user's preferred graph plotting packages and molecular visualisation tools. Fig. 2 (rendered using matplotlib) illustrates the sampling projected onto the subspace defined by the first two Principal Components (PC), PC1 and PC2, for simulations of MUP ([15], plus additional unpublished data), resolving three conformational states, while the time series of PC1 for one trajectory as shown in Fig. 3 (rendered using matplotlib) reveals how it moves between these. Finally, for a different biomolecular system, Fig. 4 illustrates (using Chimera [17]) the animation of the first PC extracted from MD simulations of a DNA tetranucleotide (sequence TGTC) [16].

In the following, an example of the use of *pyPczcomp* is shown to compare quantitatively the MUP [15] collective mode data files of the respectively free state (*apo_CA.pcz*) and bound state (*holo_CA.pcz*), using the *quick* option of pyPczcomp so as to skip the time-consuming comparisons of collective modes. This is carried out by calculating the dot product matrix between the eigenvectors identified by the PCA investigation on the apo-protein and those identified by PCA in the ligand-bound form of the protein. In particular, from this example we observe that the major similarity (63%) between the two different scenarios concerns the comparison of the first two principal components and there is a subspace overlap of 73.4%. Indeed, we would expect a degree of similarity between the two scenarios (as we would also expect differences due to ligand binding) and by means of pyPczcomp we have shown a straightforward way to quantify such similarity.

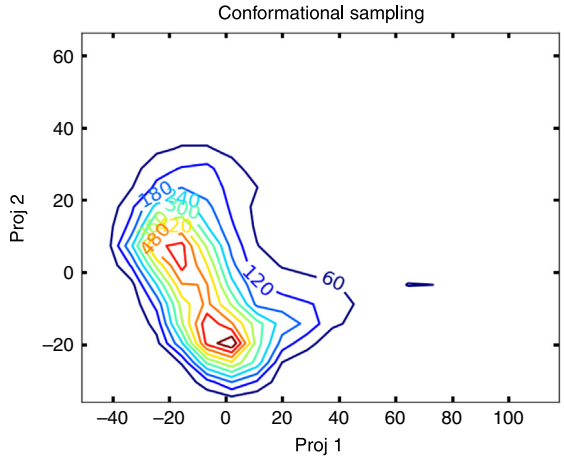

Fig. 2. Conformational sampling in MD simulations of the MUP protein, projected onto the subspace defined by the first two principal components. Two highly populated states can be distinguished, and a third, rare, state identified.

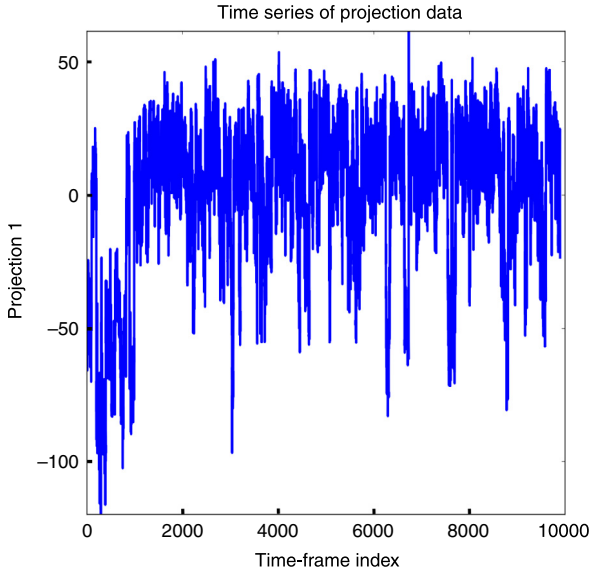

Fig. 3. Time series of the projection of snapshots from an MD simulation of MUP onto the first principal component (same data as Fig. 2). A transition from the rare state (PC1 approx. −50) to the more populated states (PC1 around 20) is seen at time frame 1000, however the rare state continues to be sampled, albeit infrequently, over the remainder of the simulation.

```
> pyPczcomp -i apo_CA.pcz holo_CA.pcz --quick
Comparison of X: apo_CA.pcz and Y: holo_CA.pcz
Rmsd between <X> and <Y>:   0.43
Dot product matrix:
        0    1    2    3    4    5    6    7    8    9
   0 0.63 0.19 0.02 0.03 0.22 0.08 0.16 0.09 0.17 0.10
   1 0.30 0.39 0.22 0.15 0.17 0.35 0.05 0.10 0.31 0.11
   2 0.22 0.03 0.03 0.40 0.15 0.11 0.08 0.06 0.05 0.29
   3 0.22 0.01 0.49 0.20 0.10 0.06 0.06 0.26 0.12 0.06
   4 0.04 0.48 0.28 0.04 0.36 0.05 0.13 0.11 0.28 0.31
   5 0.27 0.30 0.48 0.13 0.25 0.28 0.09 0.30 0.19 0.16
   6 0.11 0.09 0.30 0.43 0.18 0.56 0.07 0.18 0.01 0.13
   7 0.03 0.22 0.10 0.11 0.01 0.39 0.10 0.07 0.05 0.12
   8 0.36 0.20 0.15 0.13 0.23 0.04 0.60 0.09 0.32 0.09
   9 0.13 0.10 0.01 0.37 0.30 0.01 0.41 0.05 0.04 0.03
Subspace overlap:  0.734
Average maximum dot product:  0.464
```

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

Table 1
Compression performance.

System[a]	Atoms	90% variance		95% variance		98% variance		99% variance	
		Cmpr[b]	Error[c]	Cmpr	Error	Cmpr	Error	Cmpr	Error
1	132	0.06	1.29	0.10	0.90	0.19	0.57	0.31	0.40
2	1139	0.03	0.43	0.08	0.30	0.18	0.19	0.29	0.14
3	2494	0.10	0.42	0.17	0.30	0.27	0.20	0.33	0.15

[a] See text for details.
[b] Compression as a fraction of the original file size.
[c] Average error between original and compressed snapshots (RMSD in angstroms).

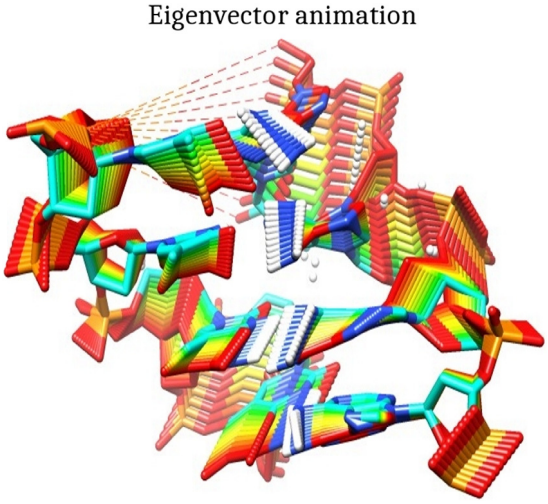

Fig. 4. Visualisation of the first principal component extracted from MD simulations of a DNA tetranucleotide (sequence TGTC). Multiple conformations of the DNA, sampling this mode, have been overlaid to demonstrate the collective motion of atoms in the structure. The colouring of carbon atoms changes from blue to red in sequential frames, while the colouring of other atoms (O, N, P, H) remains constant. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In addition to the here presented illustrative examples, an introductory tutorial that includes a variety of analysis examples that can be performed through this software, is available at https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082/blob/master/ramonbsc-pypcazip-3d7ab553c8dc/README.mdl.

6. Impact

The software presented in this paper, pyPcazip, is an easy to use, flexible and extensible package for PCA-based investigations of molecular simulation data generated by most common state-of-the-art simulation packages such as AMBER, CHARMM, GROMACS and NAMD. PCA methods are of growing importance in the Biosimulation field, as the volumes of data that can be produced using modern HPC facilities overwhelm more traditional qualitative and human operator-intensive analysis techniques. The method has been used for some time to provide key insights into the relationship between biomolecular structure, dynamics, and function. Examples of our own work include the analysis of sequence-dependent DNA dynamics [3,18–20], protein–ligand interactions [15] and GPCR dynamics [21,22] but to date there has been no open source software product designed specifically to perform this type of analysis, or compatible with all common simulation packages.

pyPcazip gives insights into structure and behaviour of molecules in addition to enabling highly compressed data storage of simulation trajectory files with insignificant loss of information. Through its analysis components the software provides a variety of methods that produce objective, quantitative and comparative metrics related to convergence and sampling of molecular simulation as well as metrics on the similarity between molecular simulation trajectories. We envisage a large potential user base for pyPcazip given that (i) PCA methods have been in use by the Biomolecular simulation community for about 15 years and that, as discussed above, (ii) the need for such automated and quantitative routes to data analysis is growing rapidly, together with the fact that (iii) the approach is applicable to trajectory data from any type of simulation (nucleic acids, proteins, oligosaccharides etc.). Moreover, pyPcazip has been promoted to the Biomolecular simulation community at several international conferences [23–27] that have contributed to the expansion of its user base. It is easy to install and use on a wide variety of different platforms ranging from personal Unix-based workstations to national HPC resources.

Finally, the source code repository of pyPcazip, together with accompanying documentation, installation instructions for a variety of platforms, testing data and examples, is distributed under the BSD license version 2. In order to increase the visibility and usage of our software package that we present in this work but also for the benefit of the user community, we have made it freely and anonymously available for download at the official Python package register (https://pypi.python.org/pypi/pyPcazip/). More than 1158 downloads in the last month were recorded at the beginning of April 2016.

7. Conclusions and future directions

pyPcazip gives insights into structure and behaviour of molecules in addition to enabling highly compressed data storage of simulation trajectory files with insignificant loss of information. It provides an easy to use, flexible approach to undertaking PCA-based investigations of MD trajectory

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

data generated by all of the most common current simulation packages.

The modularity of the software enables the integration and planning of future methodologies to complement the insights obtained from the use of the PCA method. Currently, a sparse-PCA approach that has been implemented using the main data structures of the pyPcazip package is being investigated for potential better insights on collective motions in molecular systems.

Although due to significant differences in terms of functionality (that we have mentioned in the "Problems and Background" section) and usability we could not directly compare the suite of software tools we present here with the old Fortran and/or C codes that we cite in the "Problems and Background" section, we would expect the pyPcazip performance in terms of speed to be close to the performance of Fortran and C codes given that most of the heavy algebra calculation sections in pyPcazip have been implemented via SciPy whose time-critical loops are usually implemented in C or Fortran. In addition, pyPcazip has also been designed for use on HPC architectures so that the workload can be spread across different nodes and cores of a cluster making the final results available to the user at a fraction of the time of a serial run. In addition, for the sake of even more improved performance, the replacement of the most computationally expensive routines of pyPcazip with corresponding Fortran code is being validated and an up to 10-fold improvement of performance has been observed during preliminary tests of performance analysis. We plan future software releases to incorporate these enhancements.

Finally, as an open source Python package, pyPcazip is amenable to end-user driven development and integration with the growing number of other Python-based packages in the molecular simulation domain.

Supporting material

Source code, supporting material and users support through the bitbucket ticketing system is publicly available at https://github.com/ElsevierSoftwareX/SOFTX-D-15-00082. The code is accompanied with extensive information on the application of this software, detailed installation instructions for desktop workstations but also HPC architectures and details of the code performance for differing system configurations, all available at the same public access repository.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [EP/K039490/1]; Spanish Ministry of Science and Competitiveness (MINECO; Bio2012-32868; SEV-2011-00067); National Institute of Bioinformatics (INB) PT13/0001/0019; H2020 BioExcel. This work used the ARCHER UK National Supercomputing Service. We thank many members of the CCP-BioSim network (www.ccpbiosim.ac.uk) for their input as beta-testers of the pyPcazip toolkit and I. Navarro for benchmarking pyPcazip on a Barcelona HPC cluster.

References

[1] Amadei A, Linssen AB, Berendsen HJ. Essential dynamics of proteins. Proteins 1993;17(4):412–25. http://dx.doi.org/10.1002/prot.340170408.
[2] Kitao A, Go N. Investigating protein dynamics in collective coordinate space. Curr Opin Struct Biol 1999;9(2):164–9. http://dx.doi.org/10.1016/S0959-440X(99)80023-2.
[3] Sherer EC, Harris SA, Soliva R, Orozco H, Laughton CA. Molecular dynamics studies of DNA A-tract structure and flexibility. J Am Chem Soc 1999;121:5981–91.
[4] Wlodek ST, Clark TW, Scott LR, McCammon JA. Molecular dynamics of acetylcholinesterase dimer complexed with tacrine. J Am Chem Soc 1997;119:9513–22.
[5] Howe PW. Principal components analysis of protein structure ensembles calculated using NMR data. J Biomol NMR 2001;20(1):61–70. PMID: 11430756.
[6] Yang L, Song G, Carriquiry A, Jernigan RL. Close correspondence between the essential protein motions from principal component analysis of multiple HIV-1 protease structures and elastic network modes. Structure 2008;16(2):321–30. http://dx.doi.org/10.1016/j.str.2007.12.011.
[7] Kenzaki H, Kikuchi M. Free-energy landscape of kinesin by a realistic lattice model. Proteins 2008;71(1):389–95. PMID: 17957776.
[8] Meyer T, Ferrer-Costa C, Perez A, Rueda M, Bidon-Chanal A, Luque JF, et al. Essential dynamics: A tool for efficient trajectory compression and management. J Chem Theory Comput 2006;2(2):251–8. http://dx.doi.org/10.1021/ct050285b.
[9] Laughton CA, Orozco M, Vranken W. COCO: A simple tool to enrich the representation of conformational variability in NMR structures. Proteins 2009;75:206–16.
[10] Case DA, Cheatham III TE, Darden T, Gohlke H, Luo R, Merz Jr KM, et al. The amber biomolecular simulation programs. J Comput Chem 2005;26:1668–88.
[11] Brooks BR, Brooks CL, MacKerell AD, et al. CHARMM: The biomolecular simulation program. J Comput Chem 2009;30(10):1545–614.
[12] Van Der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJ. GROMACS: fast, flexible, and free. J Comput Chem 2005;26(16):1701–18.
[13] Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, et al. Scalable molecular dynamics with NAMD. J Comput Chem 2005;26(16):1781–802.
[14] Michaud-Agrawal N1, Denning EJ, Woolf TB, Beckstein O. MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. J Comput Chem 2011;3032(10):2319–27. http://dx.doi.org/10.1002/jcc.21787.
[15] Roy J, Laughton CA. Long-Timescale molecular dynamics simulations of the major urinary protein provide atomistic interpretations of the unusual thermodynamics of ligand binding. Biophys J 2010;99(1):218–26. http://dx.doi.org/10.1016/j.bpj.2010.03.055.
[16] Pasi M, Maddocks JH, Beveridge D, Bishop TC, et al. mu ABC: a systematic microsecond molecular dynamics study of tetranucleotide sequence effects in B-DNA. Nucleic Acids Res 2014;42(19):12272–83. http://dx.doi.org/10.1093/nar/gku855.
[17] Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, et al. UCSF Chimera–a visualisation system for exploratory research and analysis. J Comput Chem 2004;25(13):1605–12. http://dx.doi.org/10.1002/jcc.20084.
[18] Harris SA, Laughton CA. A simple physical description of DNA dynamics: quasi-harmonic analysis as a route to the configurational entropy. J Phys Condens Matter 2007;19(7):076103. http://dx.doi.org/10.1088/0953-8984/19/7/076103.
[19] Harris SA, Gavathiotis E, Searle MS, Orozco M, Laughton CA. Cooperativity in drug-DNA recognition: a molecular dynamics study. J Am Chem Soc 2001;123(50):12658–63. http://dx.doi.org/10.1021/ja016233n.
[20] Harris SA, Sands ZA, Laughton CA. Molecular dynamics simulations of duplex stretching reveal the importance of entropy in determining the biomechanical properties of DNA. Biophys J 2005;88(3):1684–91. http://dx.doi.org/10.1529/biophysj.104.046912.
[21] Ng HW, Laughton CA, Doughty SW. Molecular dynamics simulations of the adenosine A2a receptor: structural stability, sampling, and convergence. J Chem Inf Model 2013;53(5):1168–78. http://dx.doi.org/10.1021/ci300610w.

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002

[22] Ng HW, Laughton CA, Doughty SW. Molecular dynamics simulations of the adenosine A2a receptor in POPC and POPE lipid bilayers: effects of membrane on protein behavior. J Chem Inf Model 2014;54(2):573–81. http://dx.doi.org/10.1021/ci400463z.

[23] Shkurti A, Laughton CA, Goni R, Bethune I, Breitmoser E, Jha S, et al. Extensible toolkit for advanced sampling and analysis in biomolecular simulation. In: 8th European conference on python in science.

[24] Shkurti A, Goni R, Orozco M, Laughton CA, New tools from the ExTASY project for the high-performance analysis of MD simulations. In: 4th annual CCP bio-sim conference: frontiers of biomolecular simulation.

[25] Shkurti A, Laughton CA, Goni R, Bethune I, Breitmoser E, Jha S, et al. Putting ExTASY in charge of an arduous computational challenge. SC14: HPC matters, New Orleans, Louisiana, 16–21 November 2014.

[26] Shkurti A, Goi R, Orozco M, Laughton CA, Towards faster-to-implement and extensible python-based tools for molecular simulation data analysis. In: 3rd annual CCP bio-sim conference: frontiers of biomolecular simulation.

[27] Shkurti A, Laughton CA, Enhanced molecular simulation analysis techniques as a core component of the ExTASY project. In: 5th international summer school on HPC challenges in computational sciences.

Please cite this article in press as: Shkurti A, et al. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.04.002